Refine Search

Search Results -

Terms	Documents
L6 and subset	12

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L7

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Friday, January 30, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; THES=ASSIGNEE; PLUR=YES; OP=OR			
L7	L6 and subset	12	L7
L6	L5 and hierarch$ and rank$ and schema$	15	L6
L5	L4 and l1	77	L5
L4	(search$ with concept$) same database	196	L4
L3	L2 and l1	16189	L3
L2	(search$ with concept$) dame database	74162	L2
L1	(macrocontext$ or context$ or "macro-context" or (macro adj context$)) and @ad<=19990803	131267	L1

END OF SEARCH HISTORY

h e b b cg b e e ch

First Hit Fwd Refs

Generate Collection Print

L7: Entry 7 of 12 File: USPT May 28, 2002

US-PAT-NO: 6397228
DOCUMENT-IDENTIFIER: US 6397228 B1

TITLE: Data enhancement techniques

DATE-ISSUED: May 28, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lamburt; Leonid	Marlboro	MA		
Koyfman; Lazar	Sudbury	MA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Verizon Laboratories Inc.	Waltham	MA			02

APPL-NO: 09/ 282342 [PALM]
DATE FILED: March 31, 1999

PARENT-CASE:
CROSS REFERENCE TO RELATED APPLICATION The present application is related to the following ten copending United States patent applications each filed on Mar. 31, 1999, each having its assignee of the entire interest in common with the assignee of the entire interest of the present application, and having titles and serial numbers as follows: TARGETED BANNER ADVERTISEMENTS, Ser. No. 09/282,764; COMMON TERM OPTIMIZATION, Ser. No. 09/282,356; GENERIC OBJECT FOR RAPID INTEGRATION OF DATA CHANGES, Ser. No. 09/283,815; ADAPTIVE PARTITIONING TECHNIQUES IN PERFORMING QUERY REQUESTS AND REQUEST ROUTING, Ser. No. 09/282,493; EFFICIENT DATA TRANSFER MECHANISM FOR SYNCHRONIZATION OF MULTI-MEDIA DATABASES, Ser. No. 09/283,816; NEW ARCHITECTURE FOR ON-LINE QUERY TOOL, Ser. No. 09/283,837; DATA MERGING TECHNIQUES, Ser. No. 09/282,295; TECHNIQUES FOR PERFORMING INCREMENTAL DATA UPDATES, Ser. No. 09/283,820; WEIGHTED TERM RANKING FOR ON-LINE QUERY TOOL, Ser. No. 09/282,730; and, HYBRID CATEGORY MAPPING FOR ON-LINE QUERY TOOL, Ser. No. 09/283,268.

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/203; 200/201
US-CL-CURRENT: 707/203; 200/201

FIELD-OF-SEARCH: 707/203, 707/200, 707/202, 707/201, 382/169, 382/124, 382/302

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

h e b b g e e e f c e e e ge

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	4003024	January 1977	Riganati et al.	382/302
☐	4365304	December 1982	Ruhman et al.	382/169
☐	5187747	February 1993	Capello et al.	382/124
☐	5802527	September 1998	Brechtel et al.	707/200
☐	6073140	June 2000	Morgan et al.	707/203

ART-UNIT: 2172

PRIMARY-EXAMINER: Shah; Sanjiv

ATTY-AGENT-FIRM: Suchyta; Leonard Charles Weixel; James K.

ABSTRACT:

Disclosed is a system for performing online data queries. The system for performing online data queries is a distributed computer system with a plurality of server nodes each fully redundant and capable of processing a user query request. Each server node includes a data query cache and other caches that may be used in performing data queries. The data query, as well as request allocation, is performed in accordance with an adaptive partitioning technique with a bias towards an initial partitioning scheme. Generic objects are created and used to represent business listings upon which the user may perform queries. Various data processing and integration techniques are included which enhance data queries. An update technique is used for synchronizing data updates as needed in updating the plurality of server nodes. A multi-media data transfer technique is used to transfer non-text or multi-media data between various components of the online query tool. Optimizations for searching, such as the common term optimization, are included for those commonly performed data queries. Also disclosed is a system for targeting advertisements that are displayed to a user of the system.

35 Claims, 71 Drawing figures

First Hit Fwd Refs

Generate Collection Print

L7: Entry 3 of 12 File: USPT Nov 19, 2002

US-PAT-NO: 6484161
DOCUMENT-IDENTIFIER: US 6484161 B1

TITLE: Method and system for performing online data queries in a distributed computer system

DATE-ISSUED: November 19, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chipalkatti; Renu	Lexington	MA		
Koyfman; Lazar	Sudbury	MA		
Getchius; Jeffrey	Cambridge	MA		
Venugopal; Ramakrishnan	Chelmsford	MA		
Scofield; Cary	Litchfield	NH		
Moratzavi; Ahmad	Sudbury	MA		
Sivasankaran; Rajendran	Waltham	MA		
Liu; Siping	Framingham	MA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Verizon Laboratories Inc.	Waltham	MA			02

APPL-NO: 09/ 283837 [PALM]
DATE FILED: March 31, 1999

PARENT-CASE:
CROSS REFERENCE TO RELATED APPLICATIONS The present application is related to the following ten copending U.S. patent applications each filed on Mar. 31, 1999, each having its assignee of the entire interest in common with the assignee of the entire interest of the present application, and having titles and serial numbers as follows: TARGETED BANNER ADVERTISEMENTS, Ser. No. 09/282,764; COMMON TERM OPTIMIZATION, Ser. No. 09/282,356; GENERIC OBJECT FOR RAPID INTEGRATION OF DATA CHANGES, Ser. No. 09/283,815; ADAPTIVE PARTITIONING TECHNIQUES IN PERFORMING QUERY REQUESTS AND REQUEST ROUTING, Ser. No. 09/282,493 now U.S. Pat. No. 6,393,415; EFFICIENT DATA TRANSFER MECHANISM FOR SYNCHRONIZATION OF MULTI-MEDIA DATABASES, Ser. No. 09/283,816 now U.S. Pat. No. 6,421,683; DATA ENHANCEMENT TECHNIQUES, Ser. No. 09/282,342 now U.S. Pat. No. 6,397,228; DATA MERGING TECHNIQUES, Ser. No. 09/282,295 now abandoned; TECHNIQUES FOR PERFORMING INCREMENTAL DATA UPDATES, Ser. No. 09/283,820; WEIGHTED TERM RANKING FOR ON-LINE QUERY TOOL, Ser. No. 09/282,730; and, HYBRID CATEGORY MAPPING FOR ON-LINE QUERY TOOL, Ser. No. 09/283,268.

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/3; 707/2, 707/4, 707/10
US-CL-CURRENT: 707/3; 707/10, 707/2, 707/4

h e b b g e e e f c e e e ge

FIELD-OF-SEARCH: 707/10, 707/6, 707/100, 707/2, 707/3, 707/4, 707/1, 707/203, 707/200, 707/201, 709/225

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL | Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5956716	September 1999	Kenner et al.	707/10
☐	6061515	May 2000	Chang et al.	707/100
☐	6253248	June 2001	Nakai et al.	707/507

ART-UNIT: 2172

PRIMARY-EXAMINER: Corrielus; Jean M.

ATTY-AGENT-FIRM: Suchyta; Leonard Charles Weixel; James K.

ABSTRACT:

Disclosed is a system for performing online data queries. The system for performing online data queries in a distributed computer system with a plurality of server nodes each fully redundant and capable of processing a user query request. Each server node includes a data query cache and other caches that may be used in performing data queries. The data query, as well as request allocation, is performed in accordance with an adaptive partitioning technique with a bias towards an initial partitioning scheme. Generic objects are created and used to represent business listings upon which the user may perform queries. Various data processing and integration techniques are included which enhance data queries. An update technique is used for synchronizing data updates as needed in updating the plurality of server nodes. A multi-media data transfer technique is used to transfer non-text or multi-media data between various components of the online query tool. Optimizations for searching, such as the common term optimization, are included for those commonly performed data queries. Also disclosed is a system for targeting advertisements that are displayed to a user of the system.

46 Claims, 71 Drawing figures

First Hit Fwd Refs

Generate Collection Print

L7: Entry 3 of 12 File: USPT Nov 19, 2002

US-PAT-NO: 6484161
DOCUMENT-IDENTIFIER: US 6484161 B1

TITLE: Method and system for performing online data queries in a distributed computer system

DATE-ISSUED: November 19, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chipalkatti; Renu	Lexington	MA		
Koyfman; Lazar	Sudbury	MA		
Getchius; Jeffrey	Cambridge	MA		
Venugopal; Ramakrishnan	Chelmsford	MA		
Scofield; Cary	Litchfield	NH		
Moratzavi; Ahmad	Sudbury	MA		
Sivasankaran; Rajendran	Waltham	MA		
Liu; Siping	Framingham	MA		

US-CL-CURRENT: 707/3; 707/10, 707/2, 707/4

ABSTRACT:

Disclosed is a system for performing online data queries. The system for performing online data queries in a distributed computer system with a plurality of server nodes each fully redundant and capable of processing a user query request. Each server node includes a data query cache and other caches that may be used in performing data queries. The data query, as well as request allocation, is performed in accordance with an adaptive partitioning technique with a bias towards an initial partitioning scheme. Generic objects are created and used to represent business listings upon which the user may perform queries. Various data processing and integration techniques are included which enhance data queries. An update technique is used for synchronizing data updates as needed in updating the plurality of server nodes. A multi-media data transfer technique is used to transfer non-text or multi-media data between various components of the online query tool. Optimizations for searching, such as the common term optimization, are included for those commonly performed data queries. Also disclosed is a system for targeting advertisements that are displayed to a user of the system.

46 Claims, 71 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 71

First Hit Fwd Refs

Generate Collection Print

L7: Entry 8 of 12 File: USPT May 21, 2002

US-PAT-NO: 6393415
DOCUMENT-IDENTIFIER: US 6393415 B1

TITLE: Adaptive partitioning techniques in performing query requests and request routing

DATE-ISSUED: May 21, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Getchius; Jeffrey	Cambridge	MA		
Scofield; Cary	Litchfield	NH		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Verizon Laboratories Inc.	Waltham	MA			02

APPL-NO: 09/ 282493 [PALM]
DATE FILED: March 31, 1999

PARENT-CASE:
CROSS REFERENCE TO RELATED APPLICATIONS The present application is related to the following ten copending United States patent applications each filed on Mar. 31, 1999, each having its assignee of the entire interest in common with the assignee entire interest of the present application, and having titles and serial numbers as follow TARGETED BANNER ADVERTISEMENTS, Ser. No. 09/282,764; now pending COMMON TERM OPTIMIZATION, Ser. No. 09/282,356; now pending GENERIC OBJECT FOR RAPID INTEGRATION OF DATA CHANGES, Ser. No. 09/283,815; now pending EFFICIENT DATA TRANSFER MECHANISM FOR SYNCHRONIZATION OF MULTI-MEDIA DATABASES, Ser. No. 09/283,816; now pending NEW ARCHITECTURE FOR ON-LINE QUERY TOOL, Ser. No. 09/283,837; now pending DATA ENHANCEMENT TECHNIQUES, Ser. No. 09/282,342; now pending DATA MERGING TECHNIQUES, Ser. No. 09/282,295; now abandoned TECHNIQUES FOR PERFORMING INCREMENTAL DATA UPDATES, Ser. No. 09/283,820; now pending WEIGHTED TERM RANKING FOR ON-LINE QUERY TOOL, Ser. No. 09/282,730; now pending and, HYBRID CATEGORY MAPPING FOR ON-LINE QUERY TOOL, Ser. No. 09/283,268 now pending.

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/2; 707/3
US-CL-CURRENT: 707/2; 707/3

FIELD-OF-SEARCH: 707/2, 707/3, 707/4, 707/1, 709/225

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

h e b b g e e e f c e e e ge

Search Selected | Search ALL | Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5898780	April 1999	Liu et al.	380/25
☐	5941947	August 1999	Brown et al.	709/225
☐	6092061	July 2000	Choy	707/1
☐	6098066	August 2000	Snow et al.	707/3
☐	6178418	January 2001	Singer	707/3

ART-UNIT: 2172

PRIMARY-EXAMINER: Shah; Sanjiv

ATTY-AGENT-FIRM: Suchyta; Leonard Charles Weixel; James K.

ABSTRACT:

Disclosed is a system for performing online data queries. The system for performing online data queries is a distributed computer system with a plurality of server nodes each fully redundant and capable of processing a user query request. Each server node includes a data query cache and other caches that may be used in performing data queries. The data query, as well as request allocation, is performed in accordance with an adaptive partitioning technique with a bias towards an initial partitioning scheme. Generic objects are created and used to represent business listings upon which the user may perform queries. Various data processing and integration techniques are included which enhance data queries. An update technique is used for synchronizing data updates as needed in updating the plurality of server nodes. A multimedia data transfer technique is used to transfer non-text or multi-media data between various components of the online query tool. Optimizations for searching, such as the common term optimization, are included for those commonly performed data queries. Also disclosed is a system for targeting advertisements that are displayed to a user of the system.

12 Claims, 71 Drawing figures

L7: Entry 8 of 12 File: USPT May 21, 2002

DOCUMENT-IDENTIFIER: US 6393415 B1
TITLE: Adaptive partitioning techniques in performing query requests and request routing

Application Filing Date (1):
19990331

Parent Case Text (2):
The present application is related to the following ten copending United States patent applications each filed on Mar. 31, 1999, each having its assignee of the entire interest in common with the assignee entire interest of the present application, and having titles and serial numbers as follow TARGETED BANNER ADVERTISEMENTS, Ser. No. 09/282,764; now pending COMMON TERM OPTIMIZATION, Ser. No. 09/282,356; now pending GENERIC OBJECT FOR RAPID INTEGRATION OF DATA CHANGES, Ser. No. 09/283,815; now pending EFFICIENT DATA TRANSFER MECHANISM FOR SYNCHRONIZATION OF MULTI-MEDIA DATABASES, Ser. No. 09/283,816; now pending NEW ARCHITECTURE FOR ON-LINE QUERY TOOL, Ser. No. 09/283,837; now pending DATA ENHANCEMENT TECHNIQUES, Ser. No. 09/282,342; now pending DATA MERGING TECHNIQUES, Ser. No. 09/282,295; now abandoned TECHNIQUES FOR PERFORMING INCREMENTAL DATA UPDATES, Ser. No. 09/283,820; now pending WEIGHTED TERM RANKING FOR ON-LINE QUERY TOOL, Ser. No. 09/282,730; now pending and, HYBRID CATEGORY MAPPING FOR ON-LINE QUERY TOOL, Ser. No. 09/283,268 now pending.

Detailed Description Text (5):
FIG. 2 depicts a Superpages Front End Server 804 which includes a varying number of server nodes 808-810 to respond to the various query requests as made by a user 800. The techniques and concepts which are described in paragraphs that follow may be used in a variety of different systems which include one or more server systems. Additionally, a single database or other datastore may be used. The techniques described herein may generally be applied to a large distributed system. Additionally, these same concepts and techniques may be applied in a single user system performing data queries and searches upon a local database.

Detailed Description Text (18):
One use of the data query cache 850, as will be described in paragraphs that follow, is its use in improving the performance in response to a user request in a subsequent query that may use a subset or superset of the data stored in the data query cache 850. A superset or composition query is one which is a boolean composite of several querying terms. A composition query may be determined by the parser 866, and the request router 854 may decide to which server node 808-810 the composition query or other query is sent for processing in accordance with domain weights as indicated in the configuration file. Reallocation of requests when a server is unavailable may be performed generally with a bias toward the initial allocation scheme as indicated also by the configuration file. There is an assumption that reallocation of a request is on a transient basis, and that the initial allocation scheme is the one to be maintained. This concept will be described in paragraphs that follow in accordance with request routing and data query caching.

Detailed Description Text (24):

h e b b g e e e f c e e e ge

The parse driver 858 generally uses a data schema description to interpret various data fields of the generic data objects. Generally, abstraction of the data interpretation into the data schema description enables different components of the parser 866 to operate upon and use generic data objects without requiring these components require code changes or recompilation in cases of the introduction of new data presentation types. Components which need to know the details of the generic data object, such as the parse driver 858, to perform certain functions, do this on a per-component basis using data schema descriptions to interpret a generic data object. This technique insulates code as included in the parser 866 from the introduction of new presentation types which may be represented as generic data objects.

Detailed Description Text (47):
Generally, the markup language files include one file or document per business for which there is an advertisement, for example, in this particular embodiment. Each of the markup language files 906 includes markup language statements, such as SGML-like statements, with tags identifying key data items in the document for each business. In this particular embodiment, the information retrieval software is Verity software which uses as input markup language files 906. Additionally, Verity uses its own schema file by which a user indicates what key words or terms as indicated in the markup language files are searchable and which of the data fields contain retrievable information. "Searchable" as used herein means fields or key words and terms upon which searches may be performed, like index searching keys. "Retrievable" as used herein generally means fields or categories with associated data that may be retrieved. All searchable fields have a tag, such as a business name or city. Identifiers are generally produced by the information retrieval software 908. Verity.TM., in this particular embodiment, produces term lists 836 in which there exists a list for each particular key word, term or category followed by a chain of identifiers that indicate the record number in the denormalized data store 904. Additionally, associated with each element in the term list which indicates a record in the denormalized data, retrievable data associated with that record may also be included. For example, if the field "zip code" includes a tag as included in the mark-up language file 906 which indicates that this particular field is searchable, it may be desired that whenever a user wishes to do a search for "zip code" what is actually retrieved or displayed to the user is the city and the state. Accordingly, in this instance, the term list and the term list data store 836 contain a list corresponding to the key word "zip code". There is a term list for each particular value of a zip code. Attached to each key word "zip code" and the particular value may be a list or a chain of identifiers. Associated with each identifier on the chain may be associated data, such as the city and state, which may be retrieved when a particular zip code is searched.

Detailed Description Text (71):
While including concepts and techniques described herein, for example, the different databases and packages commercially available which may be used, as known to those skilled in the art, vary with the type of data access using searches to be performed. In this particular embodiment, a relational database structure is used to store and retrieve information in the Front End Server 804. Other embodiments may include additional types of database storage using other commercially available packages or specialized software which facilitate each particular application.

Detailed Description Text (78):
Attributes may be added to the normalized objects, or only to a specific subset thereof. A denormalized representation of any one of the objects 402, 404, 406 contains the same number of attributes as any of the other one of the objects 402, 404, 406. This allows the denormalized objects to be transferred from the primary or secondary databases to the data manager 864 in a string format wherein each object can be identified. Accordingly, if values for a new attribute are added to only a subset of the objects, then the other objects, outside the subset, will contain a null value or some other conventional marker indicating that the

particular attribute is not defined (or contains no data) for the objects in question. For example, assume that a new attribute 420 is added. Further assume that the new attribute 420 only contains values for the object 402, but is not defined for the objects 404, 406. In that case, data space for the attribute 420 is still added to the denormalized version of the objects 404, 406, but no value is provided in the attribute 420 for the objects 404, 406.

Detailed Description Text (81):
Representing the documents (business listings) of the databases 812, 814 as generic objects facilitates modifying the documents, or a subset thereof, without modifying the parser 866. For example, if an attribute is added to some of the objects, then it is only necessary to modify the objects (schema and data) that will contain that attribute and to also modify the PHTML files 844 to include new scripting to handle that new attribute. The scripting may include statements to determine if the particular attribute exists for each object. For example, suppose the business listings were in black and white and then color was added to some of the listings. The color attribute could be added to some, but not all, of the objects only in normalized form. Once the new color attribute has been added, the denormalized versions of all of the objects would contain a data space for the attribute, but the objects that do not possess a color attribute will have a null marker. The PHTML files 844 can be modified to test if the color attribute is available in a particular object (e.g., to test for a null value) and to perform particular operations (such as displaying the color) if the attribute exists or, if the attribute does not exist for a particular object, displaying the object in black and white. In this way, the color attribute is added to some of the objects without modifying the parser 866 and without modifying existing objects that do not contain the attribute.

Detailed Description Text (85):
The technique disclosed herein relates to a new data type which abstracts the data interpretation from the data typing by using data schemas. A novel approach is the use of this data typing for rapid service deployment in search engines for advertising services on the Internet. For example, new presentation types may be introduced by an advertiser due to the large number of possible ways to present data to a user. An advertiser may wish to change the information displayed when a user performs a query that results in displaying information regarding the advertiser's business. If there are tens of thousands of advertisers which perform this task on a monthly basis, this implies a very high rate of new presentation types which an online advertising service must be able to accommodate. Use of this generic data type in GTE Superpages.TM. provides a flexible and efficient approach to incorporate these additional and new presentation types for large numbers of advertisers.

Detailed Description Text (87):
The generic data typing is optimized for performing multiple data operations by providing a small subset of possible operations or accesses upon any data of the generic data type. Therefore, these small subset of operations which are known may be optimized wherever there is a data access, for example, within the parser. This is in contrast to a non-generic data typing scheme which requires the introduction of a new data type and additional associated access patterns. In a non-generic data typing scheme there is an unlimited and unknown number of access patterns for which optimizations must be performed on an ad-hoc basis as new data types are introduced. Thus, when a new data type is introduced, the possible accesses need to be analyzed and optimized. In addition, the technique described herein provides for denormalized, flat, representations of the objects that facilitate rapid and efficient handling thereof.

Detailed Description Text (88):
The parse driver 858 uses a data schema description to interpret the various data attributes and fields of the generic data objects. Generally, the abstraction of

the data interpretation into the data schema description enables different components of the parse driver to operate upon and use generic data objects without having these components require code changes or recompilation due to the introduction of new presentation types. Components which need to know the details of the generic data object, such as the parse driver 858, to perform certain functions, do this on a per component basis by using the data schema description to interpret a generic data object. This insulates code from the introduction of new presentation types which are represented as the generic data objects.

Detailed Description Text (94):
Highly redundant caching is generally a technique that trades storage space against time by storing result sets along with subsets of these result sets. The highly redundant caching technique generally relies on the fact that the search time to locate an existing result is generally less than that amount of time which would result in creating the query result from a much larger search space.

Detailed Description Text (95):
One highly effective set manipulation technique, referred to as subsumption, is especially important in the adaption of a particular node. Subsumption is generally the derivation of query results from previous results, which can be either a superset of the requested result or subsets of the requested result. Subsumption is also the recognition of the relationship between queries and the determination of the shorted derivation path to a result set. That derivation may be the composition of several subsets resulting in a superset, or the extraction of a subset from a recognized result set. In subsumption, the presence of an additional conjunctive ("and") search term corresponds to the formation of a subset from the superset described without the additional term. The presence of an additional disjunctive ("or") search term corresponds to the identification and composition of existing subsets each described by one of the disjunctive clauses.

Detailed Description Text (96):
Consider the following example of the use of the data query cache and subsequent searches which use a subset of the data stored in the cache. For example, suppose the first request results in a query of all of the restaurants within thirty (30) miles of Boston. This query data is placed in the data query cache. A second request results in a query of all the seafood restaurants within thirty (30) miles of Boston. The second request is routed to the same node as the first request in accordance with loading configuration files, for example, as shown on FIG. 4. The second query is performed quickly by using the data query cache information and searching for a subset of the cached data indicating restaurants within thirty (30) miles of Boston for a subset of this first search data which indicates seafood restaurants. Subsequently, this second request query data which indicated all the seafood restaurants within thirty (30) miles of Boston is also stored as a separate data set within the data query cache.

Detailed Description Text (116):
It should generally be noted that in other embodiments in which other extended parentage thresholds are used, such as grandparents, the determination of the start data set in step 208 may be the data set with is closest in terms of parentage and with the least number of listings in the data set. The proximity in parentage is the primary ranking basis and the number of listings being secondary in determining ranking.

Detailed Description Text (117):
Referring now to FIG. 34, shown is a diagram of one example used in step 210 for determining and applying the best derivation sequence. In this example, the query is for Massachusetts AND RESTAURANTS AND FLOWERSHOPS. As represented in state 230, it has been determined that Massachusetts is the starting data set which is located in the data query cache. In this example, the parentage has been extended to grandparents, and Massachusetts has been determined to be the first ranking data

h e b b g e e e f c e e e ge

set in terms of parentage and number of listings in the data set. At this point, control proceeds to one of two states, 232 representing "Massachusetts AND RESTAURANTS", or 234 representing "Massachusetts AND FLOWERSHOPS". The state to which control is advanced depends generally on choosing the path with the minimum associated cost at each step. In this instance, the number of elements in the data sets "FLOWERSHOPS" (state 234) and "RESTAURANTS" (state 232) may be considered in determining cost. If the number of elements in FLOWERSHOPS is less than the number of elements in the data set RESTAURANTS, control proceeds to state 234 where each business listing in the data set FLOWERSHOP is examined to determine if it is also in Massachusetts. The resulting data set forms the set of all business listings in Massachusetts AND FLOWERSHOPS. In contrast, if the number of elements in the data set RESTAURANTS is less than FLOWERSHOPS, state 232 is entered and similar searching of the data set is performed. From either state 232 or 234, control proceeds to state 236 where searching of the data set elements is performed to produce the final resulting data set representing "Massachusetts AND RESTAURANTS AND FLOWERSHOPS". Generally, the approach just described is to advance to the next state which has the minimum cost associated until the final resulting data set is determined.

Detailed Description Text (128):
Referring now to FIG. 35, shown is a flowchart of an embodiment of the steps for forming a name associated with a data set, as may be stored in the data query cache or page cache. At step 240, a subset of query terms is determined such that a string representing a particular query is uniquely mapped to a name corresponding to a data set. In this embodiment, the subset of keys that are used in mapping a string corresponding to a query to a name of a data set include:

Detailed Description Text (131):
At step 244, a query string corresponding to a particular user query is formed using the original string as formed, for example, by the Parser of FIG. 2. The query string includes only those terms which are included in the subset as identified in step 240. If the original string does not include an item that is in the subset, for example, since the user query does not include the item as a search term, that item is omitted in forming the query string corresponding to the data set. At step 248, this query string is used to determine if a data set is located in the data query cache that corresponds to the current user query request. In this embodiment, the data sets each correspond to a filename. Thus, a lookup as to whether a data set corresponding to a particular user query exists may be determined by performing a directory lookup, for example, using file system services as may be included in an operating system upon a device which serves as a fast memory access or other caching device.

Detailed Description Text (160):
The combined search results are then sorted such that any redundant listings are removed. Any additional processing is performed, as in accordance with the user query, for example, as producing the listings which begin with "B", or only listing the top ranked fifteen (15) listings as ranked in accordance with other user specified criteria.

Detailed Description Text (163):
A variety of information retrieval techniques may be used to retrieve records stored in the Primary Database 812. Further details of the query engine 862 are presented in schematic format in FIG. 39. When the parse driver 858 of the parser 866 of one of the servers 808 delivers a parsed instruction to the query engine 862, the query engine 862 may, in an embodiment of the invention, include information retrieval software 908 to retrieve records from the Primary Database 812 that correspond to the user's query. The query engine 862 may include more than one form of information retrieval software. For example, the query engine, in addition including the information retrieval software 908 that is to be used to obtain listings in response to user queries, may further include banner ad

retrieval software 909 for retrieving advertisements that relate to the user's query.

Detailed Description Text (165):
Referring to FIG. 40, steps by which the information retrieval software 908 obtains results are set forth in a flow chart 83. The information retrieval software 908 may at a step 82 access markup language files 906, as depicted in FIG. 25, which are produced by the extraction routines 902 from the normalized data 900. In an embodiment, the markup language files consist of business listings that are stored in the Primary Database 812. The information retrieval software 908 may then, at a step 84 produce term lists 836 that are further used by the information retrieval software 908 to handle queries that are delivered to the query engine 862. The term lists 836 may consist of a linked list for each term that appears in one of the business listings, with the elements of the linked list including a document identifier for the business listing and certain statistics regarding the frequency of occurrence of the particular term in each document and in the document set as a whole. The banner ad retrieval software 909 may similarly generate and use banner ad term lists 837 that are further used by the banner ad retrieval software 909 to handle generation of appropriate banner ads. Next, at a step 90, the term lists, which in an embodiment are generated using Verity software, may be expanded at a step 86 to include synonyms for the terms appearing in the business listing. For example, if the term "diner" appears in a business listing, then the term "restaurant" might be assigned to the file for that business listing as stored in the Primary Database 812. The expansion of the listings to include synonyms of the words included in the listings may be accomplished by execution of PHTML scripts or other programming techniques. The expansion may establish a hierarchical structure; for example, the term "restaurant" may be stored in a tree that includes the subcategory of "ethnic restaurant," which may further include the sub-category "greek restaurant." PHTML scripts may be provided to establish the tree structure and to operate on the tree structure to retrieve results that will be provided to the user. The steps 82, 84 and 86 may be accomplished at initialization of the system, thus establishing and expanding the term lists 836, 837 for later use.

Detailed Description Text (166):
Once the system is initialized, the system may operate to obtain results that are to be displayed to the user. The steps for obtaining results may be seen in a flow chart 88 displayed in FIG. 41. Referring to FIG. 41, the parse driver 858 may at a step 20 parse a user query and deliver the parsed query in suitable form for handling by the query engine 862. The query engine may include the information retrieval software 908. At a step 22, the query engine 862 may operate the information retrieval software 908 to take the parsed user request and expand the query, turning the user request into a detailed query. Next, at a step 24, the information retrieval software may operate on the expanded term lists 836 by identifying documents associated with the terms identified in the expanded query. In an embodiment, the term lists 836 are the business listings described in connection with steps 82, 84 and 86 above, expanded to include synonyms and terms that are determined to be related to the words in the business listing. Identification of documents may be accomplished by a variety of information retrieval techniques. Documents may also be associated with queries by sorted relevancy ranking, clustering (automated grouping of related documents), automated document, summarization (creation of content abstracts, not simply the first few sentences of the document) and query-by-example (turning an individual document into a query in order to retrieve "more documents like this"). These functions may be accomplished by software techniques, such as having a table of pointers having as an argument a tokenized version of each possible term from the expanded user query from the step 22. The table of pointers may point to the location of a term list 836 for each such term. The term list may be a linked list of documents that include the term. The linked list may include information about each document, such as the number of occurrences of the term in the document, the inverse frequency of the term in the entire set of documents, the association of the document with other

documents, the association of the document with categories, and the like.

Detailed Description Text (168):
At a step 28 a variety of weighting algorithms can be used to rank documents identified in the step 24 according to the information stored in the term lists 836. For example, a simple weighting algorithm might take a single term query, such as a category of information, and rank each document in a term list 836 in numerical order according to the product of the term frequency (the number of times a term appears in the document) and the inverse document frequency (the inverse of the number of times the term appears in the entire document set).

Detailed Description Text (169):
Once the documents are ranked, at a step 30 a list of the ranked documents may be further processed by the information retrieval software to provide a results page. In particular, at the step 30, the information retrieval software 908 may determine categories into which the retrieved documents fall. In an embodiment, the categories are yellow pages categories, which have been previously assigned to the documents, which are business listings, prior to entry of the business listings in the Primary Database 812. Thus, at the step 30, the information retrieval software 908 determines what categories are associated with the business listings retrieved by the ranking at the step 28. Next, at a step 98, the information retrieval software 908 may compare the categories identified at the step 30 to the terms in the user query. If categories are present that do not include any of the terms in the user query, then, at a step 92, such categories may be discarded. Thus, the user will not retrieve categories that are unrelated to the user query. Such categories might otherwise appear, for example, if the information retrieval software 908 retrieves a business listing that is associated with two unrelated categories, only one of which is relevant to the user query. For example, a query for a restaurant might retrieve a listing for "Joe's restaurant and bowling alley." The information retrieval software 908 might then retrieve the categories "restaurants" and "bowling" that would have been associated with that listing. The "bowling" category would be discarded, because the user query for a restaurant is unrelated to the "bowling" category. The term comparison may use an expanded version of the terms in the query and in the categories. Thus, a category would not be discarded if it includes a synonym of a query term, even if the category does not include an exact term match.

Detailed Description Text (175):
Next, at a step 33, terms may be linked to specific contexts; that is, terms may be designated or classified as common terms in part according to their context. For example, the term "Boston," might be considered a common term if entered in the "city" field, but it might not be considered a common term if entered in a "business name" field or a "category" field. Similarly, the term "restaurant" might be a common term in the "category" field, but would not be considered a common term in the "city" field. Thus, at the step 33, the common term sets may be structured to reflect context. Thus, the bi-gram "Boston--Restaurant" might be stored as an expanded form that reflects both the term and the context in which it is to be treated as a common term, for example "City=Boston; Category=Restaurant."

Detailed Description Text (176):
Referring to FIG. 42, it may be desirable to expand, at a step 35, the terms that are to be designated as common terms. Thus, each term might be expanded to include both synonyms for the term and other terms that are semantically related to the common term in the established context for the term. For example, the common term "category=restaurant" might be expanded to cover results in which synonyms for restaurant are included in the results, such as "diner," "bar and grill," "eatery" and the like. Similarly, a city term might be expanded to include suburbs or neighborhoods; thus, the term "City=New York" would be expanded to include "City=Brooklyn," "City=Queens," and "City=Manhattan." Note that the synonyms for a given term might be different depending on the context. For example, the term

"Dorchester" might be a related term for "City=Boston," but it might not be a related term for "business name=Boston."

Detailed Description Text (177):
The pre-processing steps 32, 33 and 35 might be accomplished in a different order, and other steps might be included in embodiments of the invention. Once common terms are identified, linked to contexts, and expanded at the pre-processing steps 32, 33 and 35, it is possible to establish lists or identifiers at a step 46 that include the expanded common term n-grams. One way of dealing with common term combinations would be to generate in advance term lists 836 that are predicted to be used with some frequency (e.g., restaurants, Boston, New York, etc.) and to pre-calculate the intersection of the likely combinations. This approach requires substantial processing and would have to be performed frequently, given frequent changes in the identifiers. Instead, it is possible, at the step 46 to create special identifiers, or term lists 836, that represent the expanded common terms, as linked to their contexts. Thus, a term list 836 might consist of a linked list of documents, such as business listings, that contain the terms "Boston" and "restaurant," (or synonyms thereof) in the contexts in which those terms are common. The term lists 836 may, like other term lists 836 described elsewhere herein, may further include information as to the term frequency of each term, synonym or related term, and the inverse document frequency of the term, synonym or related term in all documents in the set. In an embodiment, the synonyms and related terms may be included in the actual business listings that are used to generate term lists 836, so that those listings will be included in the generation of common term lists. In an embodiment, the listings themselves may be classified as to common terms and synonyms or related terms of those terms. Listings may be further classified as to sub-contexts, depending on the search context. Listings using identical terms should also be included in term lists, because they use identical token identifiers for such terms. For example, the term "Boston" should be understood in a nationwide search to include listing in both Boston, Mass. and Boston, Ky., because the token for the term "Boston" will be the same in each case. Result sets must be identified as tokenwise semantically related to the classifications that are possible in a search. Results are thus classified into common term groups on a listing-by-listing basis.

Detailed Description Text (181):
A similar series of steps takes place if the user enters a query for a particular location in the city field 42 or the state field 44, or for a business name in the business name field 40. The information retrieval software 908 retrieves documents from the term lists 836 that correspond to a ranking of an expansion of the user-entered query.

Detailed Description Text (182):
When both a category and a location or a business name, or all three, are entered by the user, then the information retrieval software 908 may, in a conventional manner, retrieve term lists 836 that correspond to each of the terms of the query, such as a list corresponding to the category "restaurant" and a list corresponding to the city field "Boston." The information retrieval software 908 could then perform an intersection of the two sets and perform a ranking of the related categories (e.g., Italian restaurants in Boston, French restaurants in Boston, etc.) or related listings (for specific Boston restaurants). Because the term list 836 for documents containing the term "Boston" (including all businesses in Boston) and the term list 836 for documents containing the term "restaurant" (including all restaurants, nationwide) are both very large, the processing involved in retrieving each list and performing an intersection in order to identify matching categories or documents can be substantial. Accordingly, it is desirable to reduce the processing involved.

Detailed Description Text (183):
The information retrieval software 908 may be programmed with query rules at the

step 49 to recognize when a query includes a common term n-gram, such as "City=Boston; category=restaurant." That is, whatever common terms are identified at the pre-processing steps 32, 33 and 35 should be recognized by the information retrieval software 908, so that queries that use the common terms in the appropriate contexts (or synonyms or related terms in those contexts) are designated for special processing. In particular, the information retrieval software 908 may be programmed to execute the search for the user's query in the special area of memory that was established for storage of the special common term lists 836 at the step 48 of FIG. 42.

Detailed Description Text (198):
Referring to FIG. 46, at step 1000 a comparison is made between the phone number of an update record and the phone number field of each entry in the existing database. At step 1000, a determination is made as to whether or not the record in the latest version of the database copy is an 800 phone number. If a determination is made at step 1000 that the phone number of the current update entry is not an 800 number, control proceeds to step 1008. At step 1008, the procedure "match phone number" is performed to produce a subset of one or more entries of the existing database which match the existing phone number. Control proceeds to step 1010 where the procedure "name match" is performed. Generally, "name match" will be described in paragraphs that follow to determine whether there is a business name match for a particular entry. Control proceeds to step 1012 where "derive score" is performed based on the zip code and the name match score. Generally, the result of step 1012 produces a score representing a statistic relative to determining whether two entries in a particular database and an updated version of the database match.

Detailed Description Text (202):
If at step 1020 a determination is made that the score is less than or equal to 50%, control proceeds to step 1022. At step 1022, a determination is made as to whether or not the difference in the name length is less than or equal to three. If the difference in the name length field is not less than or equal to three, control proceeds to step 1028 where a determination is made in that no matching entry exists in the database. It should be generally be noted that the decision process and the comparison process performed in steps 1020 and 1022 are performed for each matching entry in the subset as produced from step 1008. It should generally be noted that the threshold length of three for the name length used in step 1022 may be varied and tuned for each particular embodiment and implementation.

Detailed Description Text (204):
At step 1024, the name edit distance is computed, for example, using dynamic programming techniques known to those skilled in the art, such as using a finite state machine, for each matching entry as in the subset produced by step 1008. At step 1026, if a determination is made that there are one or more entries with a distance less than 10% of the length of the update name string, then control proceeds to step 1100 of FIG. 52 where a determination is made at step 1100 as to whether or not there is only one matching entry in the subset as derived from the Step 1008.

Detailed Description Text (206):
Referring back to FIG. 46, if at step 1000 a determination is made that the phone number of the updated record is an 800 phone number, control proceeds to step 1002 where a determination is made as to whether or not the phone number, including the area code, and the zip code match one or more entries in the existing database. At step 1002, if there is a determination that one or more entries in the existing database match the phone number and zip code of the update record, control proceeds to step 1006 where a subset of one or more matching entries is found. Control then proceeds to point B indicated at step 1010 in FIG. 46 where execution continues.

Detailed Description Text (208):
Referring now to FIG. 48, shown is a flow chart of an embodiment for the "match

h e b b g e e e f c e e e ge

phone number" routine as performed at step 1008. At step 1050, a table is used with old and new area codes and exchanges to determine if there are one or more matching entries in the existing database which match the phone number of the current update entry. Generally, the processing step of 1050 and the decision made at step 1052 may be used, for example, where area codes have changed due to the increased volume of phone numbers which require additional area codes to a particular locality to be added. For example, the 508 area code may be expanded to include the 781 area code. Thus, an existing phone number may be included in the database with either the 781 or the 508 area code depending on the age of the data in the database. If a determination is made at step 1052 that either an old area code and exchange, or a new area code and exchange match, control proceeds to step 1054 where a subset of one or more matching entries is formed. Control proceeds to step 1056 where control returns to the calling procedure. In this instance, control returns to step 1008 where subsequent control proceeds to step 1010 of FIG. 46.

Detailed Description Text (210):
At step 1090, a search of the existing database is performed on the conjunction of the tokenized name field components and the zip code. Generally, the search is being performed for entries in the existing database which match zip code and the different components of the name field. At step 1092, a determination is made as to whether or not there are more than 5 matching entries in the existing database for the current update record. If at step 1092 a determination is made that there are more than five matching entries in the existing database, control proceeds to step 1094 where a determination is made that no match has been found. If at step 1092, a determination is made that there is not more than five matching entries, control proceeds to point B in the processing which is shown in FIG. 46, step 1010 where these name matching entries are used as the subset upon which subsequent processing is performed.

Detailed Description Text (211):
Referring now to FIG. 49, shown is a flow chart of the steps of one embodiment performing a "name match" as part of a routine processing as invoked from step 1010 of FIG. 46. Generally, the steps of FIG. 49 attempt to perform and find semantic equivalents of the names of a business in this particular instance. At step 1060, for each entry in the subset formed by step 1008, the name entries are canonized. Generally, canonization rules are a set of transformations which occur, for example, transforming abbreviations and the like to semantic equivalents allowing for a common denominator of terms to be searched for. For example, if all entries in a database use the entire work "incorporated" to indicate an incorporated business, then if a name entry includes the abbreviation "inc", this is expanded to the full name "incorporated" prior to being compared. Generally, the precise canonization rules or transformations depend upon the particular data being examined in a particular application.

Detailed Description Text (234):
At step 1442, redundant categories as stored by business are collapsed and detected by removing the equivalent categories. Generally, at step 1442, semantically equivalent categories are determined. Generally, this includes locating equivalent categories for which the spelling might be slightly different, or those fields which may be subsets or equivalents of other fields. For example, "animal doctor" may be interpreted as a semantic equivalent for "vet", or "veterinarian". Generally, this step may be done in an automated fashion using any programming language which is commercially available and may be used with the existing database. The technique involves dropping or not including special non-alpha-numeric characters or other words, similar to the stop words. White space may be compressed and comparison may be done on a case insensitive manner. The comparison may further be done by requiring an exact character match or with some at-a-distance technique similar to those previously described with other data processing.

Detailed Description Text (302):
If at the step 110 it is determined that no additional categories exist, then all categories to be assigned manually have been assigned, and control proceeds to a step 114, where the system returns to the first category that was not manually assigned, and it is determined whether the category will be assigned automatically based on the manual assignments. If at the step 114 it is determined that the category will be assigned automatically based on the manual assignments, then, at a step 116, the system may compare terms that appear in the category to terms that appear in each of the manually assigned categories. The system may thus obtain a ranking of the manually assigned categories in order of the degree of co-occurrence of terms. Next, at a step 118, the system may assign the same super-category as was assigned the highest-ranked of the manually assigned categories. Next, at a step 120, the system may determine whether there are any additional categories. If not, then control passes, as depicted by off-page connector B, to the flow chart 52 of FIG. 68. If additional categories remain, then control proceeds to the step 114 for the next category.

Detailed Description Text (305):
Once control has returned to the flow chart 52 of FIG. 68, meaning that all yellow pages categories have been mapped to a super-category, at a step 77 the banner ad retrieval software 909 may index the various super-categories in a banner ad term list 837. The banner ad term list 837 may take the form of a linked list of the super-categories, with each element in the list consisting of all of the terms that appear in the super-category, as well as all of the terms that appear in each of the categories that was matched to the super-category. It should be understood that these terms may be expanded, as described in connection with FIG. 40 above, so that synonyms and related terms are also stored with each super-category element. Storage of these terms may be in a hierarchical structure that is capable of execution using PHTML scripts or similar techniques.

Detailed Description Text (314):
From the table of linked lists of super-category terms established in the step 77, the banner ad retrieval software 909 may at a step 81 rank the super-categories. In particular, the system at the step 81 may rank the documents, i.e., the super-categories, according to the appearance of the words occurring in the user query and in the categories.

Detailed Description Text (315):
The ranking may be performed by a variety of techniques. One such technique obtains a number for each term that appears in the user query and in the categories that consists of the product of the term frequency for that term and the inverse document frequency for that term. The sum of all the resulting numbers may be calculated for all super-categories, and the supercategory with the highest sum may be the highest ranked document. The banner ad that was assigned to that highest ranked super-category at the step 72 of the flow chart 52 can then be displayed upon completion of the ranking step 81 of the flow chart 132.

Detailed Description Text (319):
These statistics may be further improved by weighting other factors. For example, it is possible to weight each term that appears in one of the categories that is retrieved upon execution of a user query and to normalize the IDF and RTF statistics over the weights. Thus, if a particular category deserves a higher weight, then it might be accorded higher weight in ranking super-categories. For example, a category that is manually mapped to a super-category might be given a higher weight than a category that is automatically mapped. The user query might be given a higher or lower weight, than other information. Categories with a large number of listings may be given higher weight. In an embodiment, each category is given a weight corresponding to the number of listings that are associated with the category, normalized by dividing the total number of listings. In an embodiment, the user query terms are each given a weight of one. In the weighting process, the

weight may be multiplied by the term element in performing the sum of the product of term frequency and inverse document frequency over all terms for all documents in the super-category linked list. Thus, with the weights, a normalized version of the Robertson's term frequency statistic can be obtained, permitting improved tuning of search queries beyond what is accomplished with use of the conventional Robertson's term frequency.

Detailed Description Text (320):
Upon completion of the ranking step 81, the highest ranked super-category is selected, and a banner ad that was assigned to that super-category at the step 72 of the flow chart 52 of FIG. 68 is selected. The banner ad may be retrieved, such as via a URL, from the banner ad server 809, for display to the user via the browser 824.